

Amendments to the Claims:

1. (Currently amended) A method for representing header and footer structures in a markup language document, comprising:

determining properties corresponding to a mini-document that relates to at least one section of an application document, wherein the mini-document includes a body portion, wherein the mini-document includes at least one member of a group comprising: a header and a footer;

mapping the properties of the mini-document into a markup language element that is stored with each of the markup language section properties of the application document, wherein mapping the properties includes:

setting an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,

setting a type attribute in the mini-document markup language element, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within the application document, wherein upon rendering the markup language document, the type attribute causes the body portion of the mini-document to be repeated in the application in accordance with the occurrence pattern, wherein the value is at least one member of a group comprising: an odd page value and an even page value,

setting page size properties of the application document in the section properties of the application document, wherein the page size properties includes a size value of the page, and

setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a top margin value, a bottom margin value, a left margin value, a right margin value and a position value of the location of the mini-document within the section of the application document;

mapping a type attribute that corresponds to an occurrence pattern of the body of the mini-document within the application document, wherein the type attribute causes the body portion to be repeated in the application document in accordance with the occurrence pattern indicated by the type attribute; and

storing the properties of the mini-document in the markup language document[.]; and
validating the markup language document in accordance with a schema having
definitions for the mini-document, wherein the definitions for the mini-document include a
definition for headers, a definition for footers, a definition for a context free chunk, a definition
for a paragraph element, a definition for a table element and a definition for a mini-document
type.

2-5. (Cancelled)

6. (Previously presented) The method of Claim 1, further comprising:
determining properties corresponding to an additional mini-document that relates to at
least one section of the application document;
mapping the properties of the additional mini-document into a markup language element,
wherein mapping includes mapping the properties into at least one member of a group
comprising: a context free chunk element and a table element; and
storing the properties of the additional mini-document in the markup language document.

7. (Original) The method of Claim 1, further comprising:
determining whether properties associated with all mini-documents of the application
document have been stored in the markup language document; and
processing further mini-documents when the properties associated with all mini-
documents have not been stored in the markup language document.

8. (Original) The method of Claim 1, wherein the properties of the mini-document
stored in the markup language document are understood by an application that understands the
markup language when the mini-document is not native to the application.

9. (Original) The method of Claim 1, wherein the markup language document is
manipulated on a server to substantially reproduce the mini-document of the application

document notwithstanding the presence of an application that generated the markup language document.

10. (Currently amended) A computer-readable medium for representing headers and footers in a markup language document, comprising:

determining properties relating to a mini-document used within a word-processing document, wherein the mini-document includes a body portion having text;

determining whether the mini-document is at least one member of a group comprising: a header and a footer;

writing the properties into a each of the section properties markup language elements associated with the word processing document, wherein writing the properties includes:

writing an option element in the mini-document markup language element
wherein the option element includes at least one member of a group comprising: a
header value and a footer value,

setting a type attribute, wherein the type attribute includes a value that indicates
an occurrence pattern of the body of the mini-document within the application document,
wherein upon rendering the markup language document, the type attribute causes the
body portion of the mini-document to be repeated in the application in accordance with
the occurrence pattern, and

setting a margin properties of the application document in the section properties
of the application document, wherein the margin properties include a numerical position
value of the location of the mini-document within the section of the word-processing
document;

mapping a type attribute that corresponds to an occurrence pattern of the text of the body
portion of the mini-document within the word-processing document, wherein the type attribute
causes the same text of the body portion to be repeated in the application document in
accordance with the occurrence pattern indicated by the type attribute; and

storing the properties in the markup language document such that the headers and footers of the word-processing document are substantially maintained when the markup language document is parsed by an application.

11. (Original) The computer-readable medium of Claim 10, wherein the markup language document is manipulated on a server to substantially reproduce the mini-document of the word-processing document notwithstanding the presence of an application that generated the markup language document.

12. (Original) The computer-readable medium of Claim 10, wherein the properties of the mini-document stored in the markup language document are understood by an application that understands the markup language when the mini-document is not native to the application.

13. (Cancelled)

14. (Previously presented) The computer-readable medium of Claim 10, wherein the type attribute corresponds to whether the mini-document occurs on at least one member of a group comprising: odd pages of the specified section of the application document, or even pages of the specified section of the application document.

15. (Cancelled)

16. (Previously presented) The computer-readable medium of Claim 10, further comprising:

determining properties corresponding to an additional mini-document that relates to at least one section of the word-processing document;

mapping the properties of the additional mini-document into a markup language element, wherein mapping includes mapping the properties into at least one member of a group comprising: a context free chunk element and a table element; and

storing the properties of the additional mini-document in the markup language document.

17. (Original) The computer-readable medium of Claim 10, further comprising:
determining whether properties associated with all mini-documents of the word-processing document have been stored in the markup language document; and
processing further mini-documents when the properties associated with all mini-documents have not been stored in the markup language document.

18. (Currently amended) A system for representing header and footer information in a markup language document, comprising:

a processor; and

a memory associated with computer-executable instructions configured to:

determine properties relating to a mini-document included in at least one section of an application document, wherein the mini-document includes a body portion having text;

determine whether the mini-document is at least one member of a group comprising: a header and a footer;

map the properties into a markup language element that is stored with markup language section properties of the sections of the application document, wherein mapping the properties includes:

setting an option element in the mini-document markup language element, wherein the option element includes at least one member of a group comprising: a header value and a footer value,

setting a type attribute, wherein the type attribute includes a value that indicates an occurrence pattern of the body of the mini-document within the application document, wherein upon rendering the markup language document, the type attribute causes the body portion of the mini-document to be repeated in the application in accordance with the occurrence pattern,

setting a margin properties of the application document in the section properties of the application document, wherein the margin properties include a position value of the location of the mini-document within the section of the application document, and

mapping a type attribute that corresponds to an occurrence pattern of the text of the body portion of the mini-document within the application document, wherein the type attribute causes the same text of the body portion to be repeated in the application document in accordance with the occurrence pattern indicated by the type attribute when the application document is generated from the markup language element; and

store the properties in the markup language section properties of the application document; and

a validation engine configured to validate the markup language document.

19. (Previously presented) The system of Claim 18, wherein the application is further configured to:

determine properties corresponding to an additional mini-document that relates to at least one section of the application document;

map the properties of the additional mini-document into a markup language element, wherein mapping includes mapping the properties into at least one member of a group comprising: a context free chunk element and a table element; and

store the properties of the additional mini-document in the markup language document.

20. (Original) The system of Claim 18, wherein the application is further configured to:

determine whether properties associated with all mini-documents of the application document have been stored in the markup language document; and

process further mini-documents when the properties associated with all mini-documents have not been stored in the markup language document.

21. (Original) The system of Claim 18, wherein the properties of the mini-document stored in the markup language document are understood by an additional application that understands the markup language when the mini-document is not native to the additional application.

22. (Original) The system of Claim 18, wherein the markup language document is manipulated on a server to substantially reproduce the mini-document of the application document notwithstanding the presence of the application that generated the markup language document.